

PRCS: protocole de routage pour CCN basé sur SDN

Eliau Aubry ⁽¹⁾⁽²⁾ – Thomas Silverston ⁽³⁾ – Isabelle Chrisment ⁽¹⁾⁽²⁾

⁽¹⁾ Université de Lorraine – LORIA CNRS UMR7503, 54506 Vandoeuvre-lès-Nancy, France

⁽²⁾ INRIA Nancy Grand-Est, 54506 Vandoeuvre-lès-Nancy, France

⁽³⁾ University of Tokyo – JFLI CNRS UMI3527, 7-3-1 Hongo, Bunkyo-Ku, Tokyo 113-0033, Japan

Résumé

La distribution de contenu comme la vidéo ou le partage de fichiers P2P occupe désormais la part la plus importante du trafic de l'Internet. L'architecture réseau orientée contenus (CCN) apparaît alors comme une alternative crédible au modèle actuel de l'Internet, afin de transmettre du contenu à très large-échelle. Toutefois, l'architecture CCN ne possède pas de plan de routage adapté et repose sur l'inondation du réseau ; c'est là l'un des principaux obstacles au déploiement de cette nouvelle architecture pour un Internet du Futur.

Dans cet article nous utilisons le paradigme des « réseaux logiciels » (SDN) et proposons PRCS, un nouveau protocole de routage pour l'architecture CCN. Notre solution originale s'affranchit complètement de l'architecture actuelle de l'Internet (TCP/IP) et ne repose que sur les messages CCN. Nous avons implémenté PRCS dans le simulateur NS-3 et évalué ses performances par des expériences de simulation. Nos résultats montrent que PRCS divise par 10 le nombre de messages transmis, tout en améliorant les performances des caches du réseau.

Keywords: Content-Centric Networking, Software-Defined Networking, Routing, Simulations, Performances

1 Introduction

De nouveaux services de distributions de contenus ont vu le jour sur Internet ces dernières années tels que Youtube, Netflix ou les applications P2P. Cependant, l'Internet n'a pas été conçu dans ce but et son architecture basée sur TCP/IP n'est plus adaptée à la distribution massive de contenu à très large échelle [PVA11]. Ainsi, l'architecture *Content-Centric Networking* (CCN) [JST09] a été proposée comme une alternative pour un Internet du futur. CCN se concentre sur le contenu lui-même et permet aux nœuds du réseau de conserver les données en mémoire (*caching*) pour des requêtes futures, allégeant de fait la charge du réseau et réduisant le délai pour les utilisateurs.

Bien que CCN soit largement adopté par les communautés de recherche et industriel [NDN15] [ICN15], son déploiement réel reste complexe à mettre en œuvre [YSC12] car cette architecture ne dispose pas de plan de routage adapté [YAA13]. En effet, à l'heure actuelle, les tables de commutations des nœuds sont construites en inondant le réseau de messages (*flooding*), surchargeant les ressources du réseau et empêchant de facto son déploiement à large-échelle. Il est donc nécessaire de proposer un mécanisme de routage pour CCN.

Les réseaux logiciels (*Software-Defined Networking*, SDN) [ONF12] permettent de séparer le plan de données (transmission des paquets) du plan de contrôle (décision de routage) du réseau. Ainsi, une entité programmable, le contrôleur, gère les commutateurs et routeurs de façon à ce que les nœuds deviennent des entités de commutation uniquement ; la décision de routage étant prise par le contrôleur. Les entités de commutation et le contrôleur communiquent ainsi à l'aide de protocoles de communication comme par exemple OpenFlow [MAB08] qui reposent sur la pile TCP/IP de l'Internet. Le contrôleur peut traiter les opérations de commutation au niveau des flots ou des paquets en transmettant des règles de commutation dans les nœuds.

Des travaux connexes ont été réalisés en vue d'utiliser SDN pour fournir un plan de routage pour CCN [SBD13] [NST13]. Cependant tous ces travaux reposent sur l'architecture TCP/IP, et nous pensons que ce type de solution empêche le déploiement de CCN. En effet, CCN a pour but de remplacer l'architecture actuelle de l'Internet ; si de nouveaux mécanismes pour CCN reposent sur TCP/IP, il n'y aura pas d'incitation à migrer vers une nouvelle architecture pour un Internet du futur.

Ainsi, nous proposons PRCS, un protocole de routage pour CCN basé sur le paradigme SDN. Notre solution originale ne repose plus sur l'architecture TCP/IP de l'Internet (*clean-slate approach*) afin de permettre le futur déploiement de CCN.

2 Protocole de routage basé sur SDN

2.1 Présentation de CCN

L'architecture CCN est une architecture de réseau de caches qui repose sur deux types de messages : les messages *Interest* et les messages *Data*. Chaque contenu a un nom, et les messages adressent les noms des contenus et non leurs localisations comme pour IP. Les noms sont des préfixes utilisés pour transmettre les messages d'*Interest*.

Chaque nœud CCN a trois composants principaux : Une FIB (*Forwarding Information Base*), une PIT (*Pending Interest Table*) et un CS (*Content Store*). La FIB stocke les entrées où sont retransmis les messages *Interest*, donc vers des sources de contenus potentielles. La PIT permet à un nœud de conserver une liste de messages *Interest* et l'interface sur laquelle ils ont été reçus. Les entrées de la PIT sont utilisées pour transmettre les messages *Data* sur le chemin inverse vers le nœud demandeur. Finalement, le CS stocke le contenu en mémoire du nœud, ce qui est à la base de l'architecture CCN car chacun des nœuds peut utiliser ces contenus pour satisfaire des requêtes futures.

CCN ne dispose pas de protocole de routage pour remplir la FIB des nœuds du réseau ; ce processus est basé sur l'inondation (*flooding*) du réseau quand il n'y a pas d'entrée vers où transmettre un message *Interest*. Cette inondation du réseau consomme de façon importante les ressources du réseau et ne permet donc pas son déploiement à l'échelle de l'Internet. C'est d'ailleurs la raison pour laquelle nous proposons un protocole de routage adapté à CCN et basé sur les réseaux logiciels (SDN).

2.2 Proposition PRCS

Avec SDN, le contrôleur communique avec les nœuds via des protocoles comme *OpenFlow*. Nous souhaitons proposer une approche originale n'utilisant plus IP pour adresser les nœuds et les contrôleurs. Nous utilisons ainsi les messages CCN eux-mêmes pour créer un canal de communication entre les nœuds et le contrôleur. Ensuite, notre protocole de routage PRCS (*Protocole de Routage pour CCN basé sur SDN*) se décompose en deux étapes : une étape d'amorçage (*bootstrapping*) et une de routage.

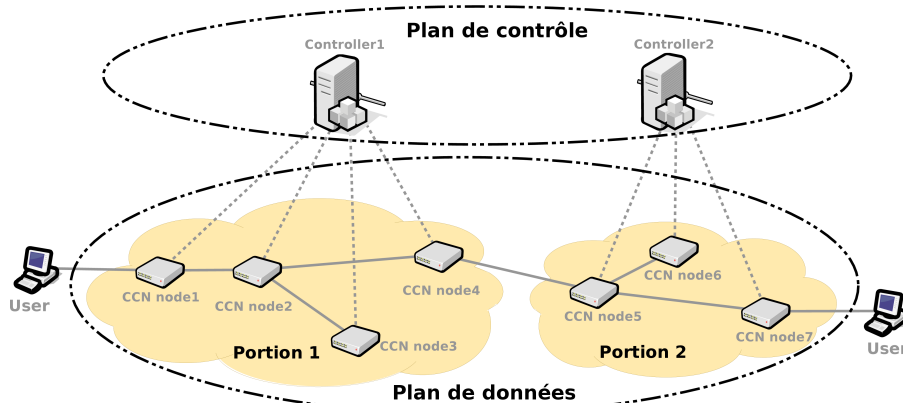


Fig. 1 – Plan de routage SDN pour architecture CCN.

Amorçage: pendant l'étape d'amorçage, le contrôleur annonce son identité aux nœuds du réseau et s'associe à eux. Ainsi, les nœuds connaissent l'interface sur laquelle relayer leurs requêtes vers leur contrôleur et cette association crée leur canal de communication. Cette étape permet au contrôleur de découvrir la topologie du réseau et d'avoir une vision globale du réseau CCN qu'il administre. Le réseau CCN peut bien sûr être administré par plusieurs contrôleurs, où chacun serait responsable d'une portion du réseau (comme illustré sur la Fig. 1). Cependant chaque nœud n'est géré que par un seul contrôleur. Dans l'étape d'amorçage, les contrôleurs détectent aussi les nœuds de bordures (nœuds 4 et 5 dans notre exemple) qui sont situés en bordure d'une portion de réseau.

Routage: l'étape de routage consiste à remplir la FIB des nœuds afin de leur permettre de relayer les messages *Interest* vers la source la plus proche (serveur original de contenu, ou Content Store d'un nœud sur le chemin), et ce, sans recourir à l'inondation du réseau. Ainsi, quand un nœud ne connaît pas l'interface sur laquelle

transmettre un message *Interest*, il interroge son contrôleur. Cet échange avec le contrôleur est effectué via le canal de communication et en utilisant les messages *Interest/Data* de CCN avec des préfixes prédéfinis. Par exemple, le message *Interest* avec le préfixe `"/controller/<idController>/path/node/<idNode>/data/<prefix>"` est envoyé au contrôleur `<idController>` par le nœud `<idNode>`, pour trouver le contenu `<prefix>`. Le contrôleur calcule le plus court chemin (ici en terme de sauts) entre le nœud et la source la plus proche et envoie la règle à tous les nœuds le long du chemin. Ainsi, les nœuds peuvent maintenant transmettre leurs requêtes (*Interest*) sur un chemin vers le contenu. Dans notre proposition, notre contrôleur a une vision globale de la topologie et de tous les contenus disponibles dans sa portion de réseau, et ce, grâce à l'étape d'amorçage. Si le contenu demandé n'est pas présent dans le réseau, le contrôleur envoie alors une règle pour atteindre les nœuds de bordure, qui vont à leur tour transmettre le message dans de nouvelles portions du réseau.

3 Environnement de Simulation

Nous avons implémenté notre protocole de routage PRCS dans le simulateur NS-3 en utilisant le module *ndnSim* [AMZ12] conçu pour simuler l'environnement CCN. Nous évaluons PRCS en le comparant à l'inondation utilisée par défaut dans CCN. Notre évaluation repose sur deux métriques principales: le nombre de messages émis dans le réseau (surcoût en communication) ainsi que les performances des *caches* du réseau (*Cache Hit*).

Les paramètres de simulation utilisés sont les suivants :

Topologie: Nous avons utilisé la topologie *Abilene* qui comprend 11 nœuds et des liens à 10Mb/s et la topologie de l'opérateur *Geant* qui compte 41 nœuds et des liens de mêmes débits.

Cache : Chaque nœud a une capacité de mémoire de 3 éléments et utilise la stratégie de cache LRU (*Last-Recently Used*) largement utilisé par CCN.

Catalogue : Le catalogue comprend 200 éléments qui sont distribués uniformément dans les nœuds du réseau.

Popularité: un utilisateur envoie 100 messages *Interest* par seconde pour récupérer du contenu. Les requêtes suivent une distribution de Zipf-Mandelbrot pour définir la popularité des contenus. Le paramètre de popularité de la distribution est $\alpha = 0.7$, une valeur communément utilisée dans ce type de modélisation. Par exemple, le catalogue de *PirateBay* est modélisé avec $\alpha = 0.75$ [AMZ12].

4 Résultats de Simulation

Les résultats de simulations sont présentés sur la Fig. 2 et sont séparés en 4 parties correspondant à l'évaluation des deux métriques (#messages, Cache Hit) avec les deux topologies (*Abilene* et *Geant*).

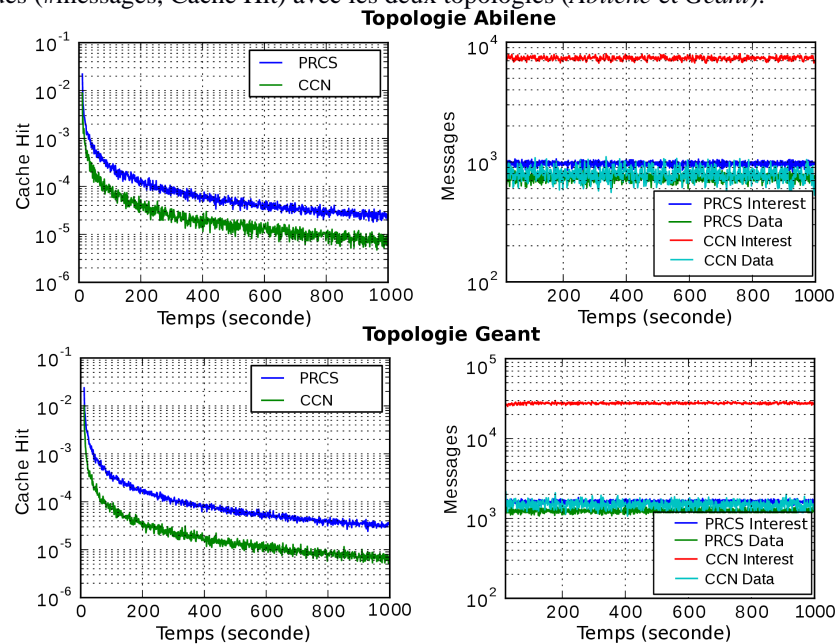


Fig. 2 – Évaluation de PRCS et comparaison à la stratégie de flooding utilisée par défaut dans CCN.

Concernant le nombre de messages, il est aisé de constater sur la figure que quelque soit la topologie utilisée, notre approche PRCS diminue par 10 le nombre de messages *Interest* envoyés dans le réseau par rapport à l'inondation utilisée par défaut dans CCN (courbe rouge et bleu foncé). Les messages de données sont bien sur équivalents puisqu'une fois localisés, les contenus sont acheminés vers leurs demandeurs.

En plus de diminuer le nombre de messages et alléger ainsi la charge du réseau, PRCS améliore également les performances des caches du réseau (*Cache Hit*), ce qui est tout à fait bénéfique pour l'architecture CCN. Il est à noter que nous avons choisi d'évaluer PRCS avec la stratégie de cache par défaut de CCN (LRU). Les performances des caches pourraient être encore améliorées si l'on utilise des stratégies de caches adaptées à CCN comme celles déjà proposées dans nos précédents travaux [CSF14].

5 Conclusion

Dans ces travaux nous avons proposé PRCS, un nouveau protocole de routage pour CCN basé sur le paradigme des réseaux logiciels (SDN). Notre proposition est une approche qui s'affranchit de l'architecture actuelle de l'Internet (TCP/IP) et n'utilise que l'architecture CCN et ses propres messages.

PRCS a été implémenté dans le simulateur NS-3 et évalué par expériences de simulation. Nos résultats montrent que PRCS est meilleur que l'approche par défaut utilisée par CCN basée sur l'inondation du réseau. En effet, PRCS diminue par 10 le nombre de messages dans le réseau tout en améliorant les performances des caches de l'architecture CCN.

PRCS est une étape importante vers le déploiement de CCN à double titre : d'une part car il répond au besoin d'un protocole de routage adapté à cette architecture, et d'autre part car il s'affranchit totalement de l'environnement TCP/IP, frein au déploiement de CCN en tant qu'architecture à la base d'un Internet du futur.

Remerciements

Ces travaux sont en partie financés par le projet ANR Doctor de l'Agence Nationale de la Recherche (ANR-14-CE28-0001) et supporté par le pôle de compétitivité "Systematic".

Références

- [AMZ12] Afanasyev A., Moiseenko I., Zhang L., '*ndnSIM: NDN simulator for NS-3*', Tech. Rep. NDN-0005, October 2012.
- [BSF14] Bernardini C., Silverston T., Festor O., '*SACS: Socially-Aware Caching Strategy for Content Centric Networking*', IFIP Networking 2014.
- [ICN15] Information-Centric Networking Research Group (ICNRG): <https://irtf.org/icnrg>
- [JST09] Jacobson V., Smetters D.K., Thornton J.D., Plass M.F., Briggs N.H., Braynard R.L., '*Networking Named Content*', ACM CoNEXT 2009.
- [MAB08] McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J., '*Openflow: Enabling Innovation in Campus Networks*', SIGCOMM Comput. Commun. Rev. 38(2):69–74, March 2008.
- [NDN15] Named-Data Project: <http://named-data.net/>
- [NST13] Nguyen X., Saucez D., Turletti T., '*Providing CCN Functionalities over OpenFlow Switches*', TR INRIA, 2013.
- [ONF12] Open Network Foundation, '*Software-Defined Networking: The New Norm for Networks*', White Paper, April 2012
- [PVA11] Perino D., Varvello M., '*A Reality Check for Content Centric Networking*', ACM SIGCOMM WS ICN 2011.
- [SBD13] Salsano S., Blefari-Melazzi N., Detti A., Morabito G., Veltri L., '*Information Centric Networking over SDN and Openflow: Architectural Aspects and Experiments on the Ofelia Testbed*', Computer Networks, 57(16):3207–3221, 2013.
- [YAA13] Yi C., Abraham J., Afanasyev A., Wang L., Zhang B., Zhang L., '*On the Role of Routing in Named Data Networking*', Tech. Rep. NDN-0016, December 2013.
- [YSC12] Yuan H., Song T., Crowley P., '*Scalable NDN Forwarding: Concepts, Issues and Principles*', IEEE ICCCN 2011.